

## CLAIMS

What is claimed is:

1. A method of processing an instruction in a processor,  
said method comprising:

fetching a sequence of instructions including  
an instruction; and

translating the instruction into separately  
executable prefetch operation and register operations,  
wherein said prefetch operation obtains, in an out-of-  
order fashion, data need to execute said register  
operation and said register operation performs an  
operation in order.

means for translating the instruction into separately executable prefetch operation and register operations, wherein said prefetch operation obtains, in an out-of-order fashion, data need to execute said register operation and said register operation performs an operation in order.

Sub  
A2

3. A processor, comprising:

a plurality of registers;

instruction processing circuitry that fetches a load instruction and a preceding instruction that precedes said load instruction in program order and, responsive to detecting said load instruction, translates said load instruction into separately executable prefetch and register operations; and

execution circuitry that performs at least said prefetch operation out-of-order with respect to said preceding instruction to prefetch data and subsequently separately executes said register operation to place said data into a register among said plurality of registers specified by said load instruction.

Cont  
A2

4. The processor of Claim 3, wherein said execution circuitry executes said register operation in-order with respect to said preceding instruction.

5. The processor of Claim 3, wherein said execution circuitry executes said register operation out-of-order with respect to said preceding instruction.

6. The processor of Claim 3, wherein said prefetch operation and said register operation have a same operation code.

7. The processor of Claim 6, wherein said prefetch operation and said register operation differ only in a value of a register operation field.

8. The processor of Claim 3, wherein said execution circuitry stores said data prefetched in response to said prefetch operation in a temporary register.

9. The processor of Claim 3, and further comprising a data hazard detector that, in response to detection of a hazard for said data, signals said processor to discard said data and said register operation.

1 Cont  
2 A2 10. A method of performing a load operation in a  
3 processor having a plurality of registers, said method  
4 comprising:

5 fetching a load instruction and a preceding  
6 instruction that precedes said load instruction in  
7 program order;

8 detecting said load instruction and translating  
9 said load instruction into separately executable prefetch  
10 and register operations;

11 performing at least said prefetch operation  
12 out-of-order with respect to said preceding instruction  
13 to prefetch data; and

14 thereafter, separately executing said register  
15 operation to place said data into a register among said  
16 plurality of registers specified by said load  
17 instruction.

1 *Cont*  
2 *A2* 11. The method of Claim 10, and further comprising  
3 executing said register operation in-order with respect  
to said preceding instruction.

1 12. The method of Claim 10, and further comprising  
2 executing said register operation out-of-order with  
3 respect to said preceding instruction.

1 13. The method of Claim 10, wherein translating said  
2 load instruction comprises translating load operation  
3 into prefetch and register operation have a same  
4 operation code.

1 14. The method of Claim 13, wherein said prefetch  
2 operation and said register operation differ only in a  
3 value of a register operation field.

1 15. The method of Claim 10, wherein performing said  
2 prefetch operation comprises storing said data in a  
3 temporary register.

1 16. The method of Claim 10, and further comprising:  
2 detecting a data hazard for said data; and  
3 in response to detection of said hazard for  
4 said data, discarding said data and said register  
5 operation.

*Add*  
*A2*